

11. Sunil

Program Name: Sunil.java

Input File: sunil.dat

Sunil has been working hard to prepare for this year's UIL programming contests. He feels fairly confident about the written test and most programming techniques but is still unsure of his skills for working with 2-dimension arrays. He wants to build a program that can perform basic processing of arrays of various sizes. Nothing complicated, just basic input and simple output to confirm proper handling of data.

Sunil has requested your assistance on this project, can you help?

Input: First line contains a single integer **T** the number of test cases that follow with $1 \leq T \leq 10$. Each test case starts with a line containing 2 integers separated by whitespace: **R**, the number of rows, and **C**, the number of columns, with both $2 \leq R, C \leq 15$. That line will be followed by **R** lines of data with each containing **C** integers separated by whitespace with integers in (-100,100).

Output: For each test case, output four lines. First line contains **R** numbers, one for each row of the array which is the average of data items in that row. Second line contains **C** numbers, one for each column of the array which is the average of data items in that column. Third line contains a single number, the average of all data items. All averages must be rounded to two decimal places and displayed right-aligned in fields that are 8 positions wide with two decimal places. The last line follows previous lines with 25 equal signs "======" starting in column 1.

Sample input:

```
3
5      2
11     -81
26     86
71     -23
-68    6
-62    48
3      6
40     90     -42    21     97     31
21     -28    -84    67     -85    -67
-30    -55    -36    -99    35     -22
7      7
38     26     -87    76     -29    -34    14
95     35     31     0      -71    -96    -99
-37    -70    -97    -39    56     -36    -27
53     -5     36     4      -36    97     -28
-14    31     17     78     -86    21     -29
16     98     99     93     -19    22     -43
6      -63    -89    99     -81    -41    -41
```

Sample output:

```
-35.00  56.00  24.00  -31.00  -7.00
-4.40   7.20
 1.40
=====
39.50  -29.33  -34.50
10.33   2.33  -54.00  -3.67  15.67  -19.33
-8.11
=====
 0.57  -15.00  -35.71  17.29   2.57  38.00  -30.00
22.43   7.43  -12.86  44.43  -38.00  -9.57  -36.14
-3.18
=====
```